Amthor, Arvid; Werner, J.; Lorenz, Andreas; Zschaeck, Stephan; Ament, Christoph:

Asymmetric motion profile planning for nanopositioning and nanomeasuring machines

URN: urn:nbn:de:gbv:ilm1-2014210268

Published OpenAccess: October 2014

Original published in:

Proceedings of the Institution of Mechanical Engineers / I. - London : Sage Publ (ISSN 2041-3041). - 224 (2010) 1, S. 79-92.
DOI: 10.1243/09596518JSCE826
URL: http://dx.doi.org/10.1243/09596518JSCE826
[Visited: 2014-10-14]

Asymmetric motion profile planning for nanopositioning and nanomeasuring machines

A Amthor, J Werner, A Lorenz, S Zschaeck and C Ament

Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering 2010 224: 79
DOI: 10.1243/09596518JSCE826

Asymmetric motion profile planning for nanopositioning and nanomeasuring machines

A Amthor*, **J Werner, A Lorenz, S Zschaeck,** and **C Ament**
System Analysis Group, Ilmenau University of Technolog, Ilmenau, Germany

The manuscript was received on 10 June 2009 and was accepted after revision for publication on 9 September 2009.

DOI: 10.1243/09596518JSCE826

Abstract: This work presents an analytic fourth-order trajectory planning algorithm, which is able to plan asymmetric motions with arbitrary initial and final velocities. Furthermore, the proposed algorithm is based on a set of quadratic derivates of jerk (djerk) functions and generates continuously differentiable trajectories in jerk, acceleration, velocity, and position under consideration of kinematic constraints in all these kinematical values. The trajectories planned by the algorithm also have time-optimal characteristics, and a synchronization between the three motion axes of the Cartesian coordinate system is ensured by the proposed method. These characteristics make it ideally suited for use as a trajectory planning algorithm in high-precision applications such as nanopositioning and nanomeasuring machines.

Keywords: analytic fourth-order trajectory generation, asymmetric motion planning, high precision motion control

1 INTRODUCTION

High-performance motion control is widely needed in modern nanopositioning. To measure and manipulate structures on the nanometre scale, high-resolution positioning stages are used. These stages are able to position a pattern in all three dimensions with an accuracy of less then one nanometre with operational ranges up to several hundred millimetres. The work presented here is motivated by a $200 \times 200\,\mathrm{mm}^2$ fine positioning stage, which was developed within the Collaborative Research Centre 622 'Nanopositioning and Nanomeasuring Machines' at the Ilmenau University of Technology [**1**, **2**]. As can be seen in Fig. 1, each axis is driven by two linear voice coil actuators. The actuators are powered by proprietary analogue amplifiers, which provide the necessary current with the required precision. Each axis is supported by two linear V-grooved guideways. The position is measured by stabilized HeNe laser interferometers with a resolution of less than 0.1 nm [**3**]. For data acquisition and control, a modular dSpace® real-time system in combination with MATLAB/Simulink® is utilized [**4**]. The control algorithm works with a sample rate of 10 kHz and operates on the amplifiers with a 16-bit resolution.

Due to the increasing operating ranges of such stages, positioning speed must also be scaled up. In recent years, trajectory tracking controllers are seeing increased usage in order to realize the requirements of a fast and simultaneously accurate motion over a wide spectrum of velocities [**1**, **5**–**7**]. As shown in Fig. 2, a trajectory tracking controller is typically composed of the following components, a trajectory planning algorithm, a feedforward controller, and a feedback controller. The task of the feedforward controller is to calculate the force which is needed to accelerate a single mass according to the desired position trajectory. To avoid a violation of the mechanical and electrical limitations of the system, the trajectory planning algorithm explicitly considers these limitations.

An important issue, especially in nanopositioning, is the stimulation of the eigenfrequencies of the mechanical components caused by abrupt changes in acceleration and jerk. These vibrations have to be minimized during motion because they decrease the accuracy and increase the settling time of the positioning stage. One way to solve this problem is

**Corresponding author: System Analysis Group, Ilmenau University of Technology, Gustav-Kirchhoff Straße 1, Ilmenau, Germany.*
email: arvid.amthor@tu-ilmenau.de

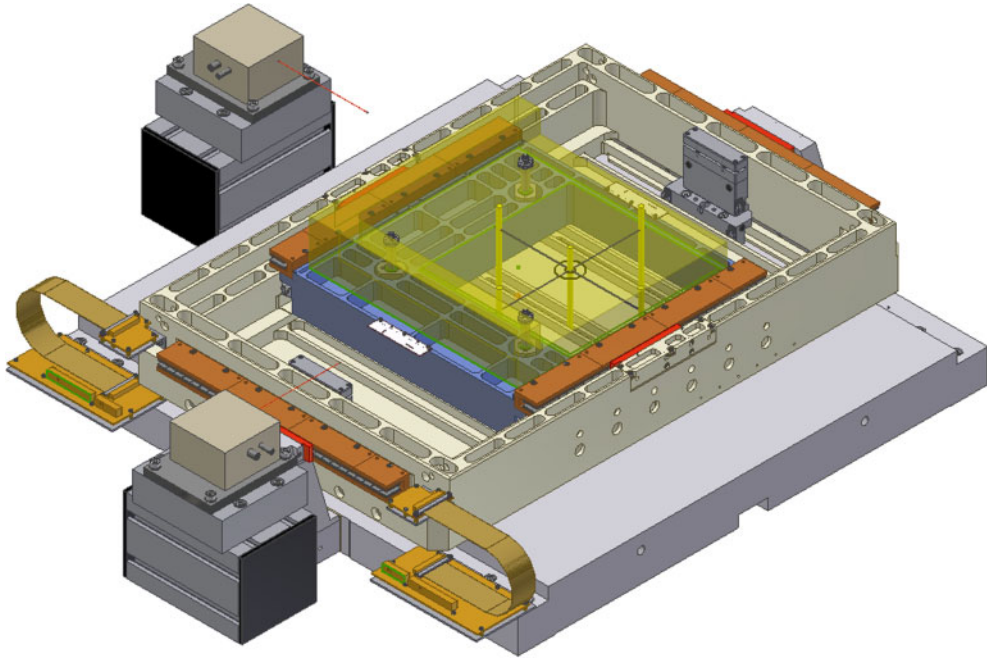

Fig. 1 XY fine positioning stage

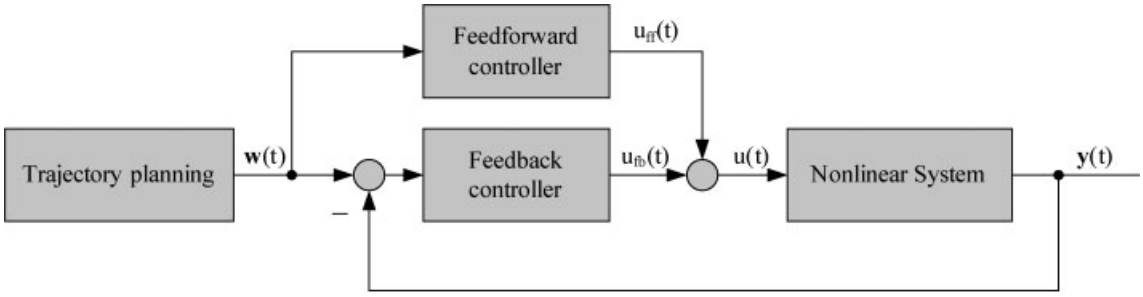

Fig. 2 Structure of a trajectory tracking controller

to generate extremely smooth motion profiles for position, velocity, acceleration, and jerk.

This paper begins by introducing the main idea of analytic trajectory planning methods. Next, the main calculation steps are described and a short overview of commonly used trajectory planning algorithms is given. Afterwards, a novel fourth-order trajectory planning algorithm, based on quadratic djerk functions, is presented. In the next section, the characteristics of the proposed trajectory planning algorithm are analysed by several simulated scenarios. Finally, several experimental results are given in order to show the effectiveness of the proposed algorithm in high-precision applications.

2 PROJECTION OF THE PLANNING TASK TO PATH LEVEL

In the last 30 years many analytical approaches have been proposed to generate smooth trajectories under consideration of kinematic constraints. In order to simplify the calculation of the set-point trajectories and to synchronize the three motion axes automatically, the proposed planning algorithms work at the one-dimensional (1D) path level [**8–11**]. The fundamental requirement for utilizing such algorithms is the possibility to define a curve in the Cartesian coordinate system against a so-called path parameter s

$$\boldsymbol{s} = \boldsymbol{s}(s) \tag{1}$$

Using such a relation, a direct mapping between the path parameter and the position on the curve is possible. This mathematical conception is demonstrated in the following. Every point $\boldsymbol{P}$ on a straight line between two points, $\boldsymbol{A}$ and $\boldsymbol{B}$, in the Cartesian coordinate system can be described with a starting point summed with a scaled normalized direction vector $\boldsymbol{\lambda}$

$$\boldsymbol{P} = \boldsymbol{A} + t\frac{(\boldsymbol{B}-\boldsymbol{A})}{\|(\boldsymbol{B}-\boldsymbol{A})\|} = A + k_{\text{path}}\boldsymbol{\lambda} \tag{2}$$

Thus, the scaling parameter k_{path} is zero at the starting point and reflects the distance between the considered points at the endpoint. This mathematical relation is the basis for the complete trajectory planning concept because at the path level it is sufficient to only consider one dimension k_{path}. Thus, the starting point can be described with $k_{\text{path}} = 0$ and the endpoint with $k_{\text{path}} = (\boldsymbol{B}-\boldsymbol{A})/\|(\boldsymbol{B}-\boldsymbol{A})\|$. Hence, equation (2) defines the considered section against the path parameter k_{path}. Using this mathematical connection a direct mapping between path level and the Cartesian coordinate system is possible.

3 PRINCIPLES OF ANALYTICAL TRAJECTORY GENERATION ALGORITHMS

The first step of an analytical trajectory generation algorithm is the mapping of the kinematic constraints to the path level. The principle idea for a section is shown schematically in Fig. 3. The constraint vector $\boldsymbol{c}_{\text{axes}}$ is rotated in the path direction (see Fig. 3(a)) and afterwards is scaled in order to fit the constraints of the motion axes (see Fig. 3(b)).

For all four constraints the valid path constraints are calculated utilizing equations (3) and (4)

$$\alpha^{\gamma}_{\min} = \min_i \left[\alpha^{\gamma}_i = \frac{c^{\gamma}_{i\,\text{axes}}}{\lambda_i}, \forall\ i = 1, \ldots, 3 \right] \tag{3}$$

$$c^{\gamma}_{\text{path}} = \alpha^{\gamma}_{\min} \times \|\boldsymbol{\lambda}\| \tag{4}$$

In equations (3) and (4), $\boldsymbol{\lambda}$ is the direction vector, $\boldsymbol{c}^{\gamma}_{\text{axes}}$ is the vector of the kinematic constraints, $\boldsymbol{c}^{\gamma}_{\text{path}}$ is the resulting vector of the constraints in path direction, and γ is the order of the derivation.

After the described constraint projection to the path level, the jerk, acceleration, velocity, and position profiles of the path parameter k_{path} are calculated in the second computation step utilizing one of the algorithms presented in sections 4 and 5.

In the final step, the kinematic profiles at the path level have to be mapped to the axes of the Cartesian coordinate system. This can be achieved by using the already explained relation between the path level and the Cartesian coordinate system may be determined. Employing the calculated kinematic trajectories of the path parameter k_{path} in equation (2), the vectors of all corresponding points in the Cartesian coordinate system may be determined. The described transformation is carried out in real-time and also works for djerk, the jerk, the acceleration, and the velocity profile.

4 STATE OF THE ART IN PLANNING KINEMATICS AT THE PATH LEVEL

This section deals with already published trajectory planning algorithms, which determine the kinematic behaviour of the path parameter k_{path} under consideration of *a priori* defined kinematic constraints. A number of third and fourth-order trajectory planning algorithms already exist, but they all make trade-offs between time optimality, computational cost, and smoothness of the generated trajectories. Furthermore, all these algorithms exhibit the commonality that they cannot create motions with an asymmetric velocity profile. The first approach in third-order trajectory planning was made by Olomski *et al.* by using a jerk trajectory with bang-bang behaviour [**9**, **10**, **12**], which is shown in Fig. 4.

The advantage of this algorithm is its ability to plan time-optimal trajectories at a marginal computational cost, but the generated acceleration profiles are unrealizable by a mechanical system.

To circumvent this problem, Sawodny *et al.* [**11**] and Sawodny *et al.* [**13**] proposed a method which uses two cubic polynomials to describe the jerk trajectory (see Fig. 5). This approach results in continuously differentiable acceleration, velocity,

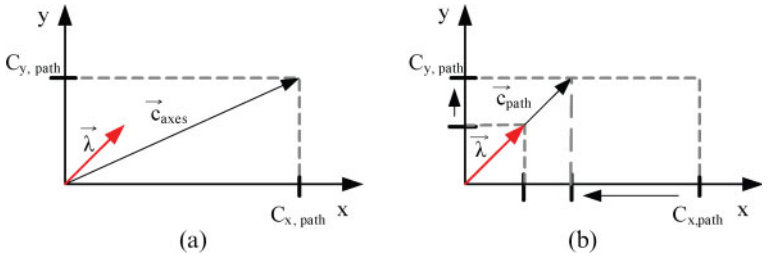

Fig. 3 Visualization of the constraint projection in two dimensions

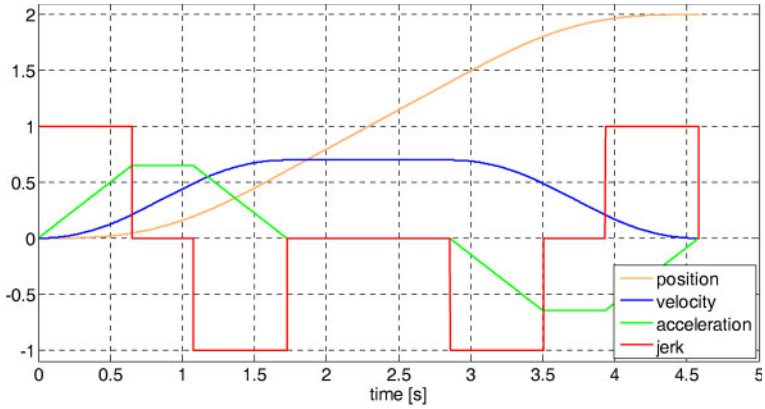

Fig. 4 Third-order trajectory generation using constant jerk functions

and position profiles, which can be realized by a mechanical system.

A comparison between Fig. 4 and Fig. 5 highlights the negative characteristic of the cubic jerk algorithm. The calculated trajectories are not time-optimal compared to the constant jerk algorithm especially in the case of badly conditioned kinematic constraints. Li *et al.* [**14**] presented a related approach using a piecewise defined sine function as the jerk profile. A deeper analysis of the algorithm has not been performed because the characteristics of the method of Li *et al.* [**14**] appear to be those of a cubic jerk algorithm. A natural extension of the described third-order trajectory planning algorithms was proposed by Lambrechts *et al.* [**8**, **15**]. They were the first to present a fourth-order motion planning algorithm, utilizing the idea of Olomski to determine all kinematic profiles. The generated trajectories for acceleration, velocity, and position are based on the profile of the djerk, and are continuously differentiable. Also, time-optimal trajectory planning is possible using this approach. Thus, Lambrechts *et al.* [**8**, **15**] merge the advantages of the constant jerk and cubic jerk methods.

5 ASYMMETRIC FOURTH-ORDER TRAJECTORY PLANNING AT THE PATH LEVEL

Due to the extraordinary requirements of nanopositioning systems the approach of Lambrechts *et al.* [**8**, **15**] is extended once again and a novel trajectory planning algorithm is presented. The main feature of the proposed algorithm is the ability to plan motions with random initial and final velocities. This provides the ability to realize continuous trajectory planning over an arbitrary number of connected path segments. The proposed algorithm is based on the construction of the djerk profile. This djerk trajectory is composed of piecewise defined quadratic functions as defined in equation (5). The trajectory can be analytically integrated several times to obtain a continuously differentiable jerk trajectory, acceleration trajectory, velocity trajectory, and position trajectory. For example, all profiles are shown in Fig. 6. Assuming a starting time of zero, the djerk trajectory can be described with seven time intervals as depicted in Fig. 6. As can be seen in equation (5), these seven time intervals define the shape of the djerk trajectory and thus all other

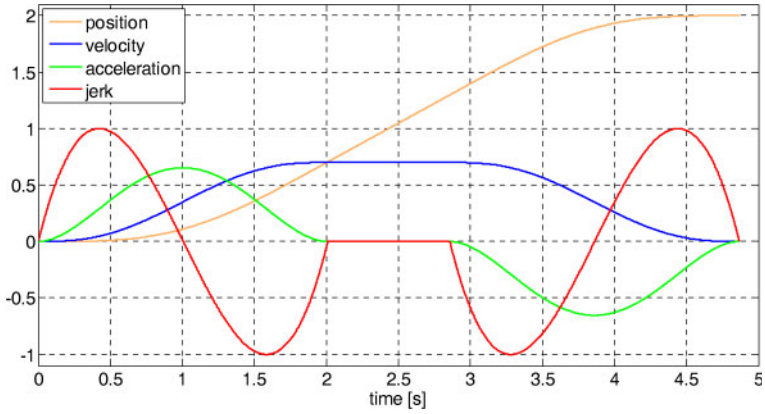

Fig. 5 Third-order trajectory generation using cubic jerk functions

kinematic trajectories. Note that the motion time is minimized if djerk, jerk, acceleration, and velocity are consecutively maximized under consideration of the given constraints.

In the following an algorithm is presented which determines these time intervals under consideration of all given kinematic constraints. In the case of an asymmetric velocity profile, all seven time intervals T_1 to T_7 are connected to each other and every interval influences the shape of all trajectories. Thus, T_1 to T_7 must be determined at the same time to ensure the fulfilment of all kinematic constraints

$$d_{\mathrm{asym}}(t)=\begin{cases}\dfrac{4d_{\max}t(T_1-t)}{T_1^2} & \text{I1}\\ 0 & \text{I2}\\ \dfrac{4d_{\max}(T_1+T_2-t)(2T_1+T_2-t)}{T_1^2} & \text{I3}\\ 0 & \text{I4}\\ \dfrac{4d_{\max}(2T_1+T_2+T_3-t)(3T_1+T_2-t)}{T_1^2} & \text{I5}\\ 0 & \text{I6}\\ -\dfrac{4d_{\max}(3T_1+2T_2+T_3-t)(4T_1+2T_2+T_3-t)}{T_1^2} & \text{I7}\\ 0 & \text{I8}\\ \dfrac{4d_{\max}(4T_1+2T_2+T_3+T_4-t)(4T_1+2T_2+T_3+T_4+T_5-t)}{T_5^2} & \text{I9}\\ 0 & \text{I10}\\ -\dfrac{4d_{\max}(4T_1+2T_2+T_3+T_4+T_5+T_6-t)(4T_1+2T_2+T_3+T_4+2T_5+T_6-t)}{T_5^2} & \text{I11}\\ 0 & \text{I12}\\ -\dfrac{4d_{\max}(4T_1+2T_2+T_3+T_4+2T_5+T_6+T_7-t)(4T_1+2T_2+T_3+T_4+3T_5+T_6+T_7-t)}{T_5^2} & \text{I13}\\ 0 & \text{I14}\\ \dfrac{4d_{\max}(4T_1+2T_2+T_3+T_4+3T_5+2T_6+T_7-t)(4T_1+2T_2+T_3+T_4+4T_5+2T_6+T_7-t)}{T_5^2} & \text{I15}\end{cases} \tag{5}$$

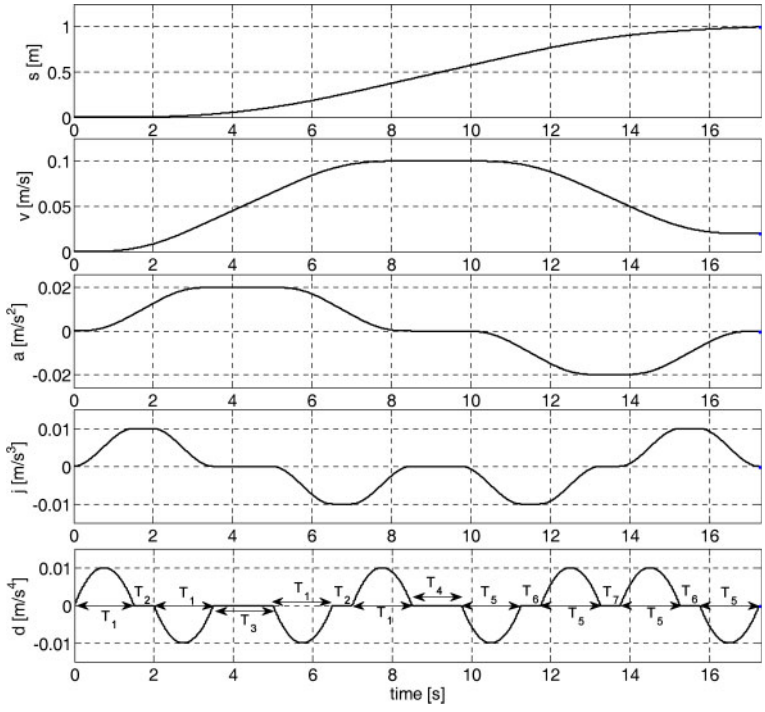

Fig. 6 Asymmetric fourth-order trajectory planning

To minimize the usage of numerical optimization algorithms, limitation of the path length is neglected in the first calculation step, and the acceleration and the breaking phase are calculated independently. The motion time is minimized if djerk, jerk, acceleration, and velocity are consecutively maximized under consideration of the given constraints. T_1 is calculated using equations (6) to (8). These times reflect the upper bound for the kinematic constraint and therefore the minimal value has to be selected (see equation (9))

$$j(t=T_1)\leqslant j_{\max} \quad \Rightarrow \quad T_{1j}\leqslant \frac{3j_{\max}}{2d_{\max}} \tag{6}$$

$$a(t=2T_1+T_2;\ T_2=0)\leqslant a_{\max} \quad \Rightarrow \quad T_{1a}\leqslant \frac{\sqrt{6d_{\max}a_{\max}}}{2d_{\max}} \tag{7}$$

$$v(t=4T_1+2T_2+T_3;\ T_2,T_3=0)\leqslant v_{\max} \quad \Rightarrow \quad T_{1v}\leqslant \frac{\sqrt[3]{6d_{\max}^2(v_{\max}-v_{\text{start}})}}{2d_{\max}} \tag{8}$$

$$T_1=\min\begin{pmatrix}T_{1j} & T_{1a} & T_{1v}\end{pmatrix} \tag{9}$$

where $d_{\max}$, $j_{\max}$, $a_{\max}$, $v_{\max}$, $s_{\max}$ are the bounds for djerk, jerk, acceleration, velocity, and path length, respectively.

A similar calculation is done for T_2 (see equations (10) to (12)) and T_3 (see equation (13)) using the previously determined interval time T_1

$$a(t=2T_1+T_2;\ T_{1\text{known}})\leqslant a_{\max} \quad \Rightarrow \quad T_{2a}\leqslant -\frac{2d_{\max}T_1^2-3a_{\max}}{2d_{\max}T_1} \tag{10}$$

$$v(t=4T_1+2T_2+T_3;\ T_3=0;\ T_{1\text{known}})\leqslant v_{\max} \quad \Rightarrow \quad T_{2v}\leqslant -\frac{3d_{\max}T_1^2-\sqrt{d_{\max}^2T_1^4+6d_{\max}T_1(v_{\max}-v_{\text{start}})}}{2d_{\max}T_1} \tag{11}$$

$$T_2=\min(T_{2a} \quad T_{2v}) \tag{12}$$

$$v(t=4T_1+2T_2+T_3;\ T_{1\text{known}},T_{2\text{known}})\leqslant v_{\max} \quad \Rightarrow \quad T_3\leqslant -\frac{4d_{\max}T_1^3+2d_{\max}T_1T_2^2+6d_{\max}T_1^2T_2+3(v_{\max}-v_{\text{start}})}{2d_{\max}T_1(T_1+T_2)} \tag{13}$$

where v_{start} describes the initial velocity. Using equations (14) to (21) T_5, T_6, and T_7 are calculated independently under the assumption that the path length is sufficient to perform the braking phase

$$j(t=T_5)\leqslant j_{\max} \quad \Rightarrow \quad T_{5j}\leqslant \frac{3j_{\max}}{2d_{\max}} \tag{14}$$

$$a(t=2T_5+T_6;\ T_6=0)\leqslant a_{\max} \quad \Rightarrow \quad T_{5a}\leqslant \frac{\sqrt{6d_{\max}a_{\max}}}{2d_{\max}} \tag{15}$$

$$v(t=4T_5+2T_6+T_7;\ T_6,T_7=0)\leqslant v_{\max} \quad \Rightarrow \quad T_{5v}\leqslant \frac{\sqrt[3]{6d_{\max}^2(v_{\max}-v_{\text{start}})}}{2d_{\max}} \tag{16}$$

$$T_5=\min\begin{pmatrix}T_{5j} & T_{5a} & T_{5v}\end{pmatrix} \tag{17}$$

$$a(t=2T_5+T_6;\ T_{5\text{known}})\leqslant a_{\max} \quad \Rightarrow \quad T_{6a}\leqslant -\frac{2d_{\max}T_5^2-3a_{\max}}{2d_{\max}T_5} \tag{18}$$

$$v(t=4T_5+2T_6+T_7;\ T_7=0;\ T_{5\text{known}})\leqslant v_{\max} \quad \Rightarrow \quad T_{6v}\leqslant -\frac{3d_{\max}T_5^2-\sqrt{d_{\max}^2T_5^4+6d_{\max}T_5(v_{\max}-v_{\text{start}})}}{2d_{\max}T_5} \tag{19}$$

$$T_6=\min(T_{2a} \quad T_{2v}) \tag{20}$$

$$v(t=4T_5+2T_6+T_7;\ T_{5\text{known}},T_{6\text{known}})\leqslant v_{\max} \quad \Rightarrow \quad T_7\leqslant -\frac{4d_{\max}T_5^3+2d_{\max}T_5T_6^2+6d_{\max}T_5^2T_6+3(v_{\max}-v_{\text{end}})}{2d_{\max}T_5(T_5+T_6)} \tag{21}$$

where v_{end} is the final velocity at the end of the path. The last calculation step is the determination of the constant velocity phase according to equation (22) by the path length, this time is kept and the other (not limited) time is replaced in equations (23) and (24). This procedure is continued until interval times

$$
\begin{aligned}
&v(t=4T_1+2T_2+T_3+T_4+4T_5+2T_6+T_7;\ T_{1\text{known}}, T_{2\text{known}}, T_{3\text{known}}, T_{5\text{known}}, T_{6\text{known}}, T_{7\text{known}}) \leqslant s_{\max}\\
&\Rightarrow \quad T_4 \leqslant -\frac{1}{6d_{\max}T_1^2T_2+2d_{\max}T_1T_2T_3+4d_{\max}T_1^3+2d_{\max}T_1T_2^2+2d_{\max}T_1^2T_3}\\
&\quad\times\big(3d_{\max}T_3T_1T_2^2-d_{\max}T_5T_6T_7^2+2d_{\max}T_1T_2T_3T_7+4d_{\max}T_1T_6T_2^2+24d_{\max}T_1^2T_2T_5\\
&\quad-d_{\max}T_1^2T_3^3-3d_{\max}T_5T_6^2T_7+10d_{\max}T_1^2T_2^2+12d_{\max}T_1^2T_2T_6+9d_{\max}T_1^2T_2T_3\\
&\quad-9d_{\max}T_5^2T_6T_7+2d_{\max}T_1T_2^2T_7-d_{\max}T_5^2T_7^2+16d_{\max}T_1^3T_2+8d_{\max}T_1^2T_3T_5\\
&\quad+4d_{\max}T_1^3T_7+8d_{\max}T_1T_2^2T_5+6d_{\max}T_1^2T_2T_7+6T_2v_{\text{start}}+3T_3v_{\text{start}}6T_6v_{\text{start}}\\
&\quad+3T_7v_{\text{start}}+12T_1v_{\text{start}}+6d_{\max}T_1^3T_3+2d_{\max}T_1T_2^3+4d_{\max}T_1^2T_3T_6+12T_5v_{\text{start}}\\
&\quad+2d_{\max}T_1^2T_3-8d_{\max}T_5^4-10d_{\max}T_5^2T_6^3+8d_{\max}T_1T_2T_3T_5+8d_{\max}T_1^3T_6-2d_{\max}T_5T_6^3\\
&\quad+4d_{\max}T_1T_2T_3T_6-6d_{\max}T_5^3T_7-2s_{\max}+8d_{\max}T_1^4+d_{\max}T_1T_2T_3^2-16d_{\max}T_5^3T_6\\
&\quad+16d_{\max}T_1^3T_5\big)
\end{aligned}
\tag{22}
$$

If $T_4 > 0$, the path length is long enough and the analytical determined switching times are valid. For the case where T_4 has a negative value, the path length is the limiting condition. Thus, all calculated time intervals are not valid and T_1 to T_7 are directly connected to each other. Under the assumption that T_1 and T_5 are limited by the path length the equations (23) and (24) have to be fulfilled

$$
\begin{aligned}
&s(t=4T_1+2T_2+T_3+T_4+4T_5+2T_6\\
&\quad+T_7;\ T_2, T_3, T_4, T_6, T_7=0)=s_{\text{end}}\\
&\Rightarrow \quad 8d_{\max}T_1^4+16d_{\max}T_1^3T_5+12v_{\text{start}}T_1\\
&+12v_{\text{start}}T_5-8d_{\max}T_5^4-3s_{\text{end}}=0
\end{aligned}
\tag{23}
$$

$$
\begin{aligned}
&v(t=4T_1+2T_2+T_3+T_4+4T_5+2T_6\\
&\quad+T_7;\ T_2, T_3, T_4, T_6, T_7=0)=v_{\text{end}}\\
&\Rightarrow \quad 4d_{\max}T_1^3-4d_{\max}T_5^3+3(v_{\text{start}}-v_{\text{end}})=0
\end{aligned}
\tag{24}
$$

Using these equations, $T_{1\text{new}}$ and $T_{5\text{new}}$ can be analytically determined. If both times are smaller than the previously calculated times $T_{1\text{old}}$ and $T_{5\text{old}}$, the assumption that T_1 and T_5 are limited by the path length is correct. In the case of a larger $T_{1\text{new}}$ and $T_{5\text{new}}$, T_2 and T_6 are possibly limited by the path length and equations (23) and (24) are utilized in a modified form (all times apart from T_2 and T_6 are now zero) in order to check this premise. In the case that only one of the times $T_{1\text{new}}$ and $T_{5\text{new}}$ is limited in the acceleration and deceleration phase are found, which are limited by the path length (namely both times are smaller than the previously analytically calculated interval times). For clarification, Fig. 7 shows the algorithm which determines all interval times if $T_4 < 0$.

With the determination of T_1 to T_7, all parameters of the djerk trajectory are available and thus all other kinematic trajectories of the path parameter k_{path} are defined.

In the last step of the planning algorithm, the kinematic profiles at the path level must be mapped to the motion axes using the already explained relation between the path level and the Cartesian coordinate system (see equation (2)).

The main advantage of this approach is the avoidance of an explicit synchronization between the motion axes, because of the geometric correlation between the path level and the Cartesian coordinate system.

It should be mentioned, that in the case of $v_{\text{start}} = v_{\text{end}}$, a completely analytical determination of the interval times T_1 to T_7 is possible, because the motion has a symmetric character. Hence, the whole profile is determined by four time intervals: the constant djerk interval, the constant jerk interval, the constant acceleration interval, and the constant velocity interval. Using this relation all set-point trajectories are symmetrical and $T_1 = T_5$, $T_2 = T_6$ as well as $T_3 = T_7$. Now the planning algorithm has to determine only four time intervals (T_1 to T_4) instead of seven.

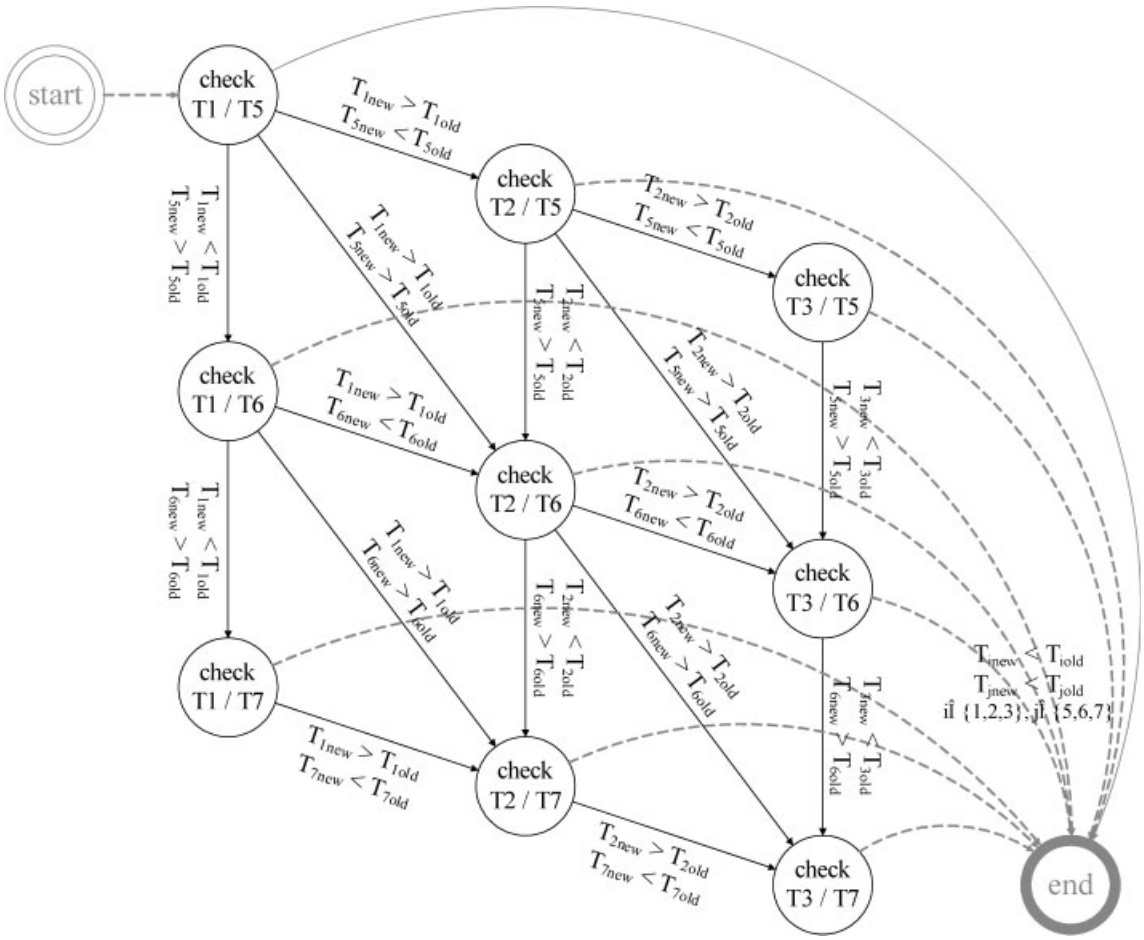

Fig. 7 Visualization of the algorithm for determination of the time intervals T_1 to T_7 if $T_4 < 0$

In the symmetrical case, T_1 is calculated as already described (see equations (6) and (7)) but now the limitation of the position can be included in the calculation because the acceleration and the braking phase have the same shape. Thus, the calculation is extended to include equations (25) to (27)

$$v(t = 4T_1 + 2T_2 + T_3;\ T_2, T_3 = 0) \leqslant v_{\max}$$
$$\Rightarrow \quad T_{1v} \leqslant \frac{\sqrt[3]{6d_{\max}^2 v_{\max}}}{2d_{\max}} \tag{25}$$

$$s(t = 8T_1 + 4T_2 + 2T_3 + T_4;\ T_2, T_3, T_4 = 0) \leqslant s_{\max}$$
$$\Rightarrow \quad T_{1s} \leqslant \frac{\sqrt[4]{3d_{\max}^3 s_{\max}}}{2d_{\max}} \tag{26}$$

$$T_1 = \min\left(T_{1j} \quad T_{1a} \quad T_{1v} \quad T_{1s}\right) \tag{27}$$

A similar calculation is done for T_2 (see equations (10) and (28) to (30)) and T_3 (see equations (31) to (33)) using the previously determined interval times

$$v(t = 4T_1 + 2T_2 + T_3;\ T_3 = 0;\ T_{1\text{known}}) \leqslant v_{\max}$$
$$\Rightarrow \quad T_{2v} \leqslant -\frac{3d_{\max}T_1^2 - \sqrt{d_{\max}^2 T_1^4 + 6d_{\max}^2 T_1 v_{\max}}}{2d_{\max}T_1} \tag{28}$$

$$s(t = 8T_1 + 4T_2 + 2T_3 + T_4;\ T_3, T_4 = 0;\ T_{1\text{known}}) \leqslant s_{\max}$$
$$\Rightarrow \quad T_{2s} \leqslant \frac{\sqrt[3]{d_{\max}^2 T_1^4 \left(8d_{\max}T_1^4 + 81s_{\max} + 9\sqrt{s_{\max}\left(16d_{\max}T_1^4 + 81s_{\max}\right)}\right)}}{6d_{\max}T_1} + \frac{2d_{\max}T_1^3}{3\sqrt[3]{d_{\max}^2 T_1^4 \left(8d_{\max}T_1^4 + 81s_{\max} + 9\sqrt{s_{\max}\left(16d_{\max}T_1^4 + 81s_{\max}\right)}\right)}} - \frac{5}{3}T_1 \tag{29}$$

$$T_2 = \min(T_{2a} \quad T_{2v} \quad T_{2s}) \tag{30}$$

$$v(t = 4T_1 + 2T_2 + T_3;\ T_{1\text{known}};\ T_{2\text{known}}) \leqslant v_{\max} \quad \Rightarrow \quad T_{3v} \leqslant -\frac{4d_{\max}T_1^3 + 6d_{\max}T_1^2T_2 + 2d_{\max}T_1T_2^2 + 3v_{\max}}{2d_{\max}T_1(T_1 + T_2)} \tag{31}$$

$$\begin{aligned} &s(t = 8T_1 + 4T_2 + 2T_3 + T_4;\ T_4 = 0;\ T_{1\text{known}}, T_{2\text{known}}) \leqslant s_{\max} \\ &\Rightarrow \quad T_{3s} \leqslant -\left(6d_{\max}T_1^3 + 3d_{\max}T_1T_2^2 + 9d_{\max}T_1^2T_2 \right. \\ &\qquad \left. - \sqrt{\begin{array}{l} 4d_{\max}^2T_1^6 + 13d_{\max}^2T_1^4T_2^2 + 12d_{\max}^2T_1^5T_2 + d_{\max}^2T_1^2T_2^4 \\ + 6d_{\max}^2T_1^3T_2^3 + 6d_{\max}T_1T_2s_{\max} + 6d_{\max}T_1^2s_{\max} \end{array}}\right) \Big/ 2d_{\max}T_1(T_1 + T_2) \end{aligned} \tag{32}$$

$$T_3 = \min(T_{3v},\ T_{3s}) \tag{33}$$

The determination of T_1, T_2, and T_3 is followed by the calculation of T_4. This interval is only bounded by the maximum path length and so T_4 is calculated by

$$\begin{aligned} &s(t = 8T_1 + 4T_2 + 2T_3 + T_4;\ T_{1\text{known}}, T_{2\text{known}}, T_{3\text{known}}) \leqslant s_{\max} \\ &\Rightarrow \quad T_4 \leqslant -\frac{1}{2d_{\max}T_1\left(T_2^2 + T_1T_3 + T_2T_3 + 2T_1^2 + 3T_1T_2\right)} \times \left(16d_{\max}T_1^4 + 2d_{\max}T_1^2T_3^2 + 18d_{\max}T_1^2T_2T_3 \right. \\ &\qquad \left. + 32d_{\max}T_1^3T_2 + 4d_{\max}T_1T_2^3 + 12d_{\max}T_1^3T_3 + 2d_{\max}T_1T_2T_3^2 + 6d_{\max}T_1T_2^2T_3 + 20d_{\max}T_1^2T_2^2 - 3s_{\max}\right) \end{aligned} \tag{34}$$

The shown analytical determination of all interval times speeds up the calculation time significantly and therefore the implementation of the proposed algorithm uses an analytical method of calculation as often as possible.

6 IMPLEMENTATION ASPECTS AND PERFORMANCE ANALYSIS

The trajectory generation algorithm is implemented in MATLAB/Simulink and consists of two modules. The determination of the valid constraints (see equations (3) and (4)) at the path level (see section 3), and the calculation of the interval times T_1 to T_7 (see section 5) are implemented in MATLAB m-code. The generation of the trajectories in djerk, jerk, acceleration, velocity, and position for the path parameter k_{path} and the subsequent coordinate transformation in the Cartesian coordinate system are done using a Simulink C-code-s-function. The reason for using C-code-s-functions is that this part of the trajectory planning algorithm has to be carried out on a real-time system. Most of these systems were programmed in C and therefore a hand-tuned implementation will achieve the best performance. Furthermore, C-code-s-functions allow the possibility to recycle the programmed algorithms.

In the following several examples for symmetric and asymmetric trajectories are provided and it will be shown that the proposed algorithm is able to reproduce the time optimality of the constant jerk method as well as the continuously differentiable jerk trajectories of the algorithm proposed by Sawodny *et al.* [**11**, **13**]. In Fig. 8, the maximum djerk is quite high in comparison to the constraints of the residual kinematic values and thus the behaviour of the constant jerk method can be imitated by the proposed algorithm. As can be seen, the planned trajectory at the path level is time-optimal; however, the calculated jerk trajectory is continuously differentiable and thus jumps in jerk can be avoided. Also, an emulation of the cubic jerk method is possible if the maximum djerk is in the same range in comparison to the other kinematic constraints (see Fig. 9). This analysis shows that the maximum djerk is a powerful tuning parameter, enabling the possibility to regulate the time optimality of the planned trajectory. Figure 10 shows a comparison between the cubic jerk method and the proposed algorithm to clarify the capability to generate time-optimal and extremely smooth trajectories. It can be easily seen that the algorithm is able to plan a continuously differentiable jerk trajectory in connection with constant jerk intervals. This leads to a reduction of the moving time by about 25 per cent compared to the cubic jerk method. All the presented examples show clearly that the proposed algorithm

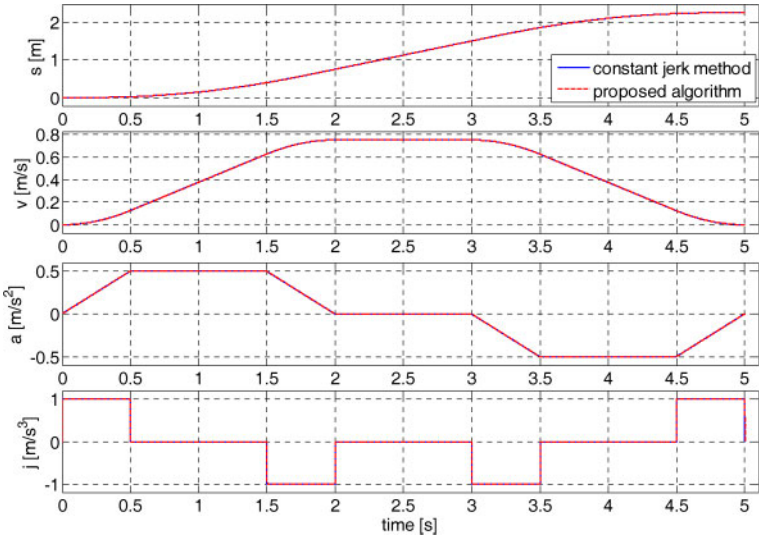

Fig. 8 Constant jerk method reproduced by the proposed algorithm

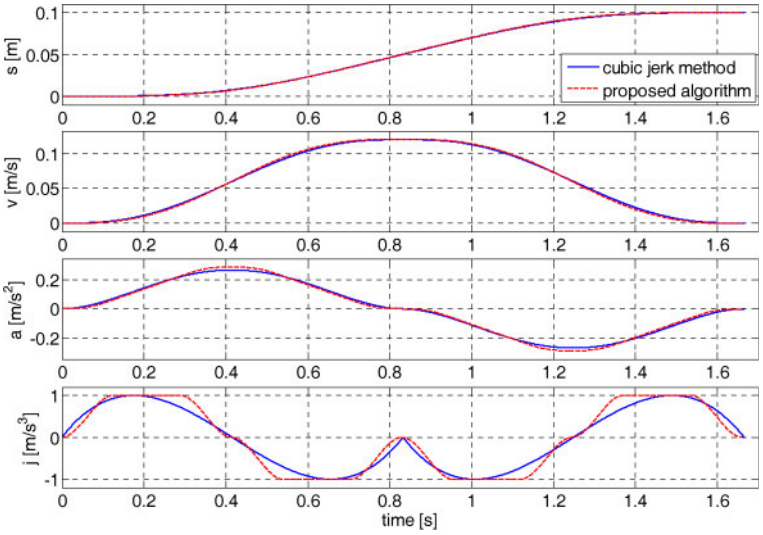

Fig. 9 Cubic jerk method reproduced by the proposed algorithm

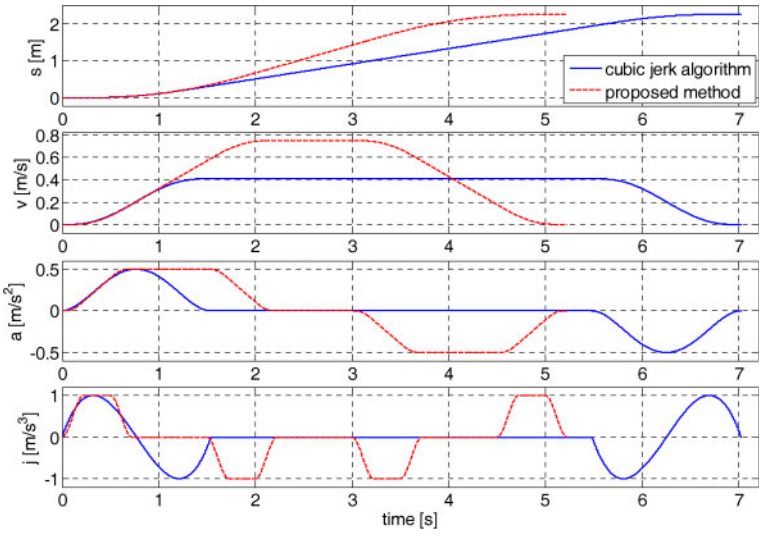

Fig. 10 Dynamical performance of the proposed algorithm compared with the cubic jerk method

unifies the advantages of the constant jerk and the cubic jerk methods.

In Fig. 11, an asymmetric motion is shown. The motion starts with an initial velocity and in the following 2 s the system is accelerated to the maximum allowed velocity. After 3 s the system begins to decelerate in order to reach the defined final velocity at the defined position.

Using the proposed approach, it is possible to plan a continuous motion along a complex trajectory in three dimensions, which is composed of several linear segments.

7 EXPERIMENTAL RESULTS

In order to verify the practical use of the proposed trajectory generation method, the algorithm was carried out on the modular dSpace real-time system described in section 1. As already mentioned the trajectory generation works with a sampling rate of 10 kHz and requires 5 µs for computation (using a 3 GHz AMD Opteron). Figure 12 shows a typical linear motion of the *y*-axis over a distance of 30 mm. As can easily be seen, the kinematic characteristic constraints of nanopositioning machines (NPMs)

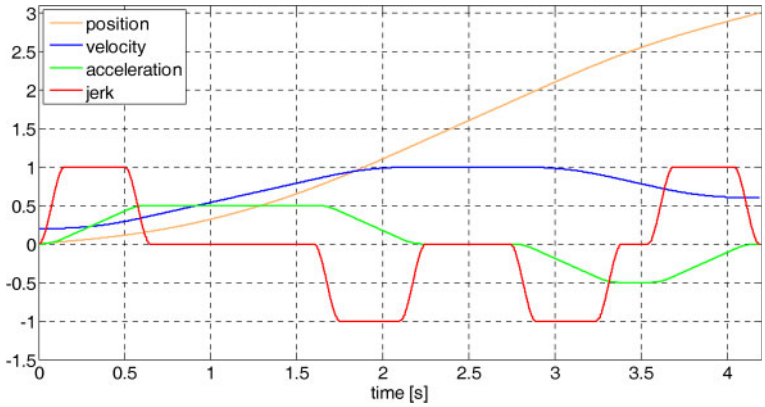

Fig. 11 Example for a movement with arbitrary initial and final velocity

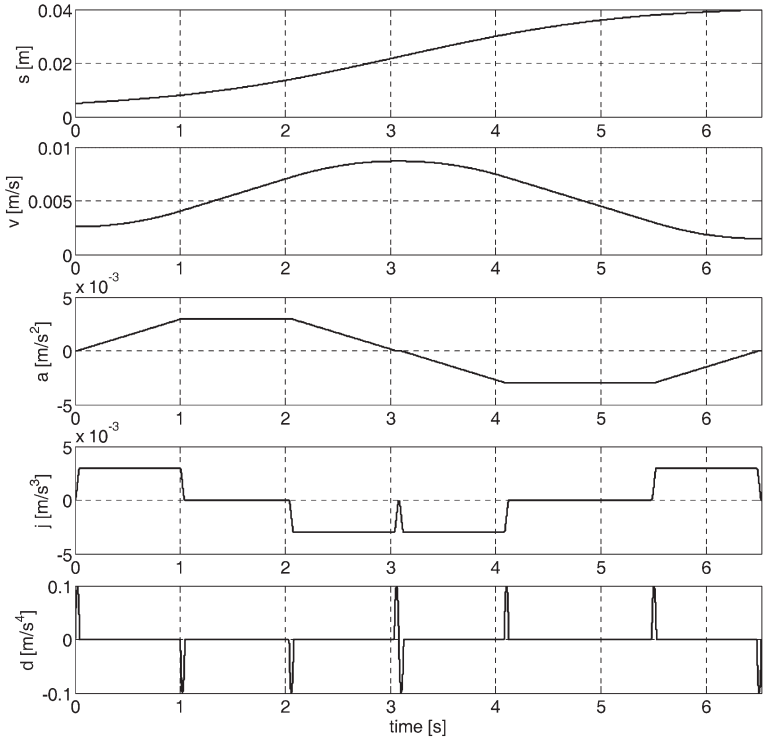

Fig. 12 Linear motion of the *y*-axis with an asymmetric velocity profile

were kept and all generated motion profiles up to jerk level are continuously differentiable.

Furthermore, it has to be mentioned, that in conventional applications the interval time T_1 only lasts for a small number of sample periods and this leads to significant discretization errors in the djerk profile. When considering high-precision applications such as NPM systems, however, this diagnosis is not correct because the sample rate is quite high compared to classical applications in robotics. To support this conclusion Fig. 13 shows the generated djerk profile from the experiment presented in Fig. 12. It can be seen, that T_1 last for nearly 45 ms, corresponding to approximately 450 sample periods. Thus, the discretization error of the djerk profile is insignificant if the sample rate is high enough. In addition to the already described experiment, another test was carried out to demonstrate the ability of the proposed algorithm to enhance the dynamic performance of the controlling system. Figures 14 and 15 show two runs of a linear motion of the *y*-axis over a distance of 30 mm, each with a different planning algorithm.

In both runs the position was controlled by a well-tuned classical proportional–integral–derivative controller in combination with a trajectory planning

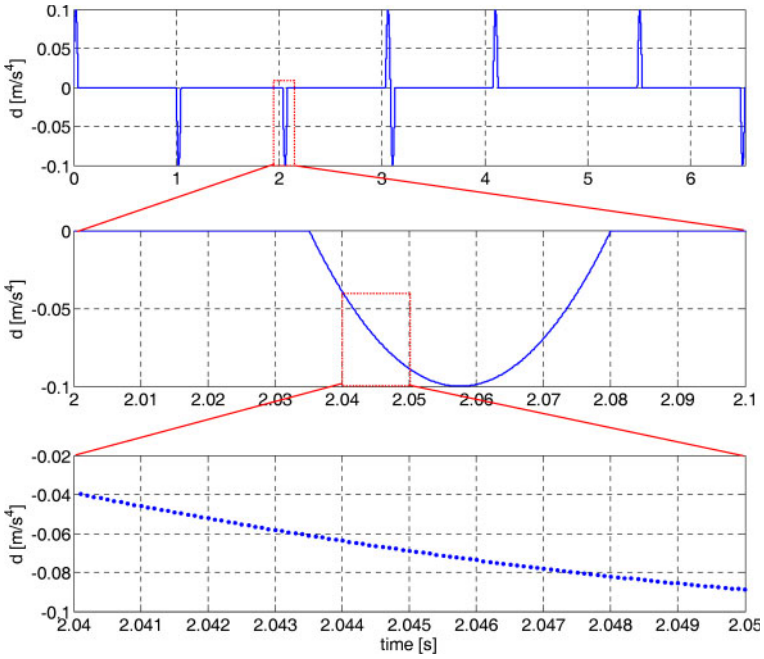

Fig. 13 Discretization of the profile at the djerk level

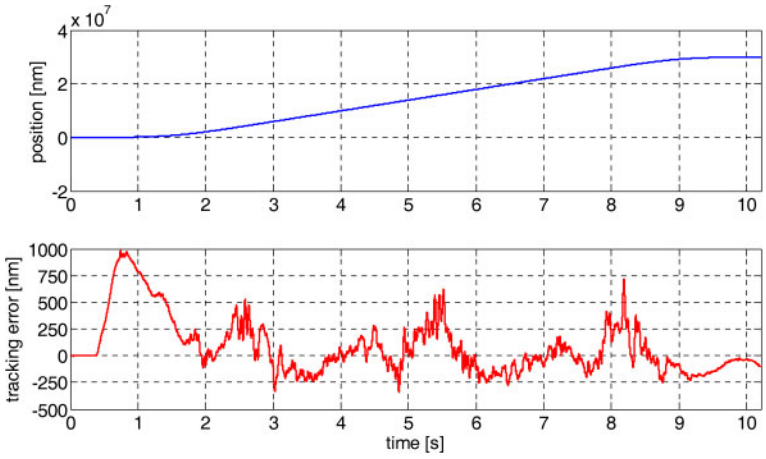

Fig. 14 Tracking error using the constant jerk method for trajectory generation

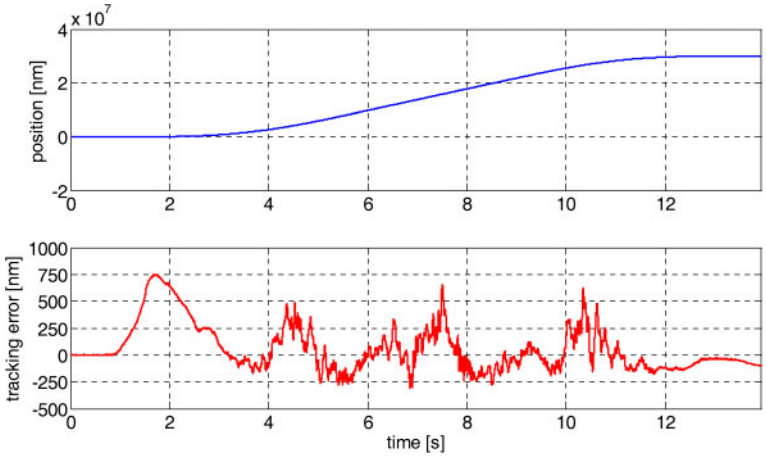

Fig. 15 Tracking error using the proposed method for trajectory generation

algorithm. In Fig. 14 the constant jerk method was utilized (see section 4). It can be seen that especially at the starting point the tracking error is quite high at almost 1000 nm. In Fig. 15 the proposed fourth-order trajectory planning algorithm is used.

The tracking error in the acceleration phase is reduced by about 25 per cent to approximately 750 nm; this clearly shows the effectiveness of the proposed algorithm in high-precision applications such as nanopositioning.

8 CONCLUSIONS AND FUTURE WORK

In this contribution, a novel trajectory planning algorithm is shown. The basis of the proposed method is the transformation of a linear segment in the three-dimensional (3D) Cartesian coordinate system to the so-called 1D path level. This is achieved by using the scaling vector of the segments' vector notation as the path parameter. In the first calculation step the kinematic constraints in djerk, jerk, acceleration, and velocity for every axis of the Cartesian coordinate system are mapped to the path level. This is followed by the second step; determination of the kinematic behaviour of the path parameter. All kinematic trajectories are based on the djerk profile, which is composed of piecewise-defined quadratic functions. This trajectory is analytically integrated several times to obtain the trajectories for jerk, acceleration, velocity, and position. The last step of the proposed method is the projection of the planned 1D trajectories to the 3D Cartesian coordinate system. It was shown by simulation and experiment, that by using this method it is possible to create continuously differentiable set-point trajectories up to the jerk level. Furthermore, the fourth-order approach offers the ability to plan motions with arbitrary initial and final velocities; this new functionality will be the basis for future work.

The next milestone will be the assembly of an arbitrary set of motion segments, which can be passed through without stopping. In order to realize this objective, the segments have to be connected with special transition curves such as clothoid or Bloss curves, because a system following the curve at constant speed will have a constant rate of angular acceleration (clothoid and Bloss curve) and a constant rate of angular jerk (Bloss curve), respectively. Furthermore, these curves cannot be analytically integrated and so a numeric integration method has to be found, which enables the possibility to integrate the transition curves in real-time with accuracy beyond a picometre. Another problem that arises is the determination of the transition velocities between the motion segments in order to give the system enough space for braking, especially if the last segment is very short compared to the segments before. Finally, all mentioned future work has to be experimentally tested with the fine positioning stage described in section 1.

ACKNOWLEDGEMENTS

The work was done within the framework of the Collaborative Research Centre 'Nanopositioning and Nanomeasuring Machines' at Ilmenau University of Technology, which is supported by the German Research Foundation (DFG) and the Thuringian Ministry of Science. The authors would also like to thank all colleagues who offered help with the work presented.

REFERENCES

1 **Amthor, A., Zschaeck, S.,** and **Ament, C.** Position control on nanometer scale based on an adaptive friction compensation scheme. In Proceedings of The 34th Annual Conference of the IEEE Industrial Electronics Society, Orlando, 10–13 November 2008, pp. 2568–2573 (IEEE Xplore).

2 **Müller, M., Amthor, A., Fengler, W.,** and **Ament, C.** Model-driven development and multi-processor implementation of a dynamic control algorithm for nanopositioning and nanomeasuring machines. *J. Syst. Control Engng*, 2009, **223**(1), 417–429.

3 SIOS Meßtechnik GmbH. 29 October 2009, Available at http://www.SIOS.de.

4 Workbench for the Nano World (customer application) dSPACE GmbH. March 2009, Available from http://www.dspace.de.

5 **Amthor, A., Hausotte, T., Ament, C., Li, P.,** and **Jäger, G.** Friction identification and compensation on nanometerscale. In Proceedings of the *IFAC World Congress 2008*, Seoul, Korea, 6–11 July 2008, pp. 2014–2019 (Elsevier).

6 **Amthor, A., Hausotte, T., Li, P.,** and **Jäger, G.** Friction modelling on nanometerscale and experimental verification. In Proceedings of the *Internationales wissenschaftliches kolloquium, volume 1*, Ilmenau, Germany, 10–13 September 2007, pp. 495–501 (The Rector of Ilmenau TU).

7 **Amthor, A., Zschaeck, S.,** and **Ament, C.** Adaptive reibkraftkompensation zur modellbasierten positionsregelung von nanopositionier- und nanomessmaschinen. *at-Automatisierungstechnik*, 2009, **59**(2), 51–59 (in German).

8 **Lambrechts, P., Boerlage, M.,** and **Steinbuch, M.** Trajectory planning and feedforward design for electromechanical motion systems. *Control Engng Pract.*, 2004, **13**(3), 145–157.

9 **Olomski, J., Rathjen, O.,** and **Leonhard, W.** Continuous path generation of reference trajectories with limited jerk and nonlinear feedforward-feedback control of industrial robots. In Proceedings of the International Workshop on *Robotics*, Madrid, Spain, 1987, pp. 63–70.

10 **Olomski, J.** Trajectory planning, optimization and control for industrial robots. In Proceedings of the International Conference on *Control and applications*, Jerusalem, Israel, 3–6 April 1989 (IEEE Xplore).

11 **Sawodny, O., Aschemann, H., Lahres, S.,** and **Hofer, E. P.** A low cost material handling and logistic system for shop floor manufacturing using an automated bridge crane. In Proceedings of the 14th *IFAC world congress*, Beijing, People's Republic of China, 5–9 July 1999, pp. 517–522 (Pergamon).

12 **Leonhard, W.** Trajectory control of a multi-axes robot with electrical servo drives. *IEE Control Sys. Mag.*, 1990, **10**(6), 3–9.

13 **Sawodny, O., Aschemann, H.,** and **Lahres, S.** An automated gantry crane as a large workspace robot. *Control Engng Pract.*, 2002, **10**(12), 1323–1338.

14 **Li, H. Z., Gong, Z. M., Lin, W.,** and **Lippa, T.** Motion profile planning for reduced jerk and vibration residuals. *SIMTech Tech. Rep.*, 2007, **8**(1), 32–37.

15 **Lambrechts, P., Boerlage, M.,** and **Steinbuch, M.** Trajectory planning and feedforward design for high performance motion systems. In Proceedings of the *American control conference*, Boston, MA, 30 June–2 July 2004, pp. 4637–4642 (American Automatic Control Council).

APPENDIX

Notation

a	acceleration on path level
$a_{\max}$	maximum acceleration on path level
$\boldsymbol{A}$	start point in Cartesian coordinates
$\boldsymbol{B}$	end point in Cartesian coordinates
$\boldsymbol{c}_{\text{axes}}$	constraint vector in Cartesian coordinates
$\boldsymbol{c}_{\text{path}}$	constraint vector on path level
d	derivative of jerk on path level
$d_{\max}$	maximum derivative of jerk on path level
j	jerk on path level
$j_{\max}$	maximum jerk on path level
k	scaling parameter
$\boldsymbol{P}$	arbitrary point in Cartesian coordinates
s	path length or path parameter
$s_{\max}$	maximum path length
t	time
T_x	length of the time intervals
v	velocity on path level
$v_{\max}$	maximum velocity on path level
v_{start}	initial velocity at the beginning of the movement
v_{end}	final velocity at the end of the movement
$\alpha_{\min}^{\gamma}$	scaling parameter of the constraint vector at the path level
γ	order of the deviation
$\boldsymbol{\lambda}$	normalized direction vector in Cartesian coordinates